In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

sns.set_style('whitegrid')

import warnings
warnings.filterwarnings('ignore')
```

In [32]:
```
analysing_sales=pd.read_csv('MOHAMMED VI UNIVERSITY SCHOOL SHOP SALES.csv')
```

In [33]:
```
analysing_sales.head()
```

Out[33]:

	DAYS	WOODY	CRESTED VEST	T-SHIRT	LONG SLEEVES	SALES(DHS)	Unnamed: 6	Unnamed: 7
0	1	400	500	340	590	1831	NaN	NaN
1	2	460	670	987	543	2662	NaN	NaN
2	3	500	900	300	345	2048	NaN	NaN
3	4	870	960	450	780	3064	NaN	NaN
4	5	200	1000	490	980	2675	NaN	NaN

In [48]:
```
Analysing_sales_after_drop=analysing_sales.drop(['Unnamed: 6', 'Unnamed: 7'], 1)
```

In [49]:
```
Analysing_sales_after_drop.head()
```

Out[49]:

	DAYS	WOODY	CRESTED VEST	T-SHIRT	LONG SLEEVES	SALES(DHS)
0	1	400	500	340	590	1831
1	2	460	670	987	543	2662
2	3	500	900	300	345	2048
3	4	870	960	450	780	3064
4	5	200	1000	490	980	2675

In [50]:
```
Analysing_sales_after_drop.describe
```

Out[50]:
```
<bound method NDFrame.describe of     DAYS  WOODY  CRESTED VEST  T-SHIRT  LONG SLEEVES  SALES(DHS)
0      1    400           500      340           590         1831
1      2    460           670      987           543         2662
2      3    500           900      300           345         2048
3      4    870           960      450           780         3064
4      5    200          1000      490           980         2675
5      6    340           300      400           659         1705
6      7    235           300      500           470         1512
7      8    280           768      700           870         2626
8      9    430           560      548           650         2197
9     10    230           435      658           560         1893
10    11    480           450      280           460         1681
11    12    670           320      345           280         1627
12    13    439           657      780           599         2488
13    14    450           769      230           435         1898
14    15    580           457      658           760         2470
15    16    280           659      437           780         2172
16    17    500           500      768           560         2345
17    18    540           659      432           560         2209
18    19    340           650      769           790         2568
19    20    280           567      890           800         2557
20    21    500           658      679           543         2401
21    22    700           700      700           700         2822
22    23    439           765      789           547         2563
23    24    230           907      768           654         2583
24    25    900           900      900           888         3613
25    26    657           678      654           567         2582
26    27    654           655      543           764         2643
27    28    432           987      670           200         2317
28    29    457           765      987           800         3038
29    30    380           890      657           450         2407
30    31    345           679      655           987         2697
31    32    675           879       98           760         2444
32    33    456           432      876           790         2587
33    34    657           890      765           345         2691
34    35    333           654      790           765         2577
35    36    455           768      908          1000         3167
36    37   2356           760     1235           766         5154
37    38    123           555      580           435         1731
38    39    789           768      456           555         2607
39    40    654           345      567           768         2374
40    41    234           654      980           780         2689
41    42    646           565      389           876         2518
42    43    540           670     1290           444         2987
43    44    459           780      980           670         2933
44    45    500           600      700           690         2535
45    46    500           500      500           500         2046
46    47    464           235      657           450         1853
47    48    769           879      450           450         2596
48    49    540           988      870           800         3247
49    50    200           368      989           677         2284>
```

In [51]:
```
Analysing_sales_after_drop.shape
```

Out[51]:
```
(50, 6)
```

In [52]:
```
Analysing_sales_after_drop.columns
```

Out[52]:
```
Index(['DAYS', 'WOODY', 'CRESTED VEST', 'T-SHIRT', 'LONG SLEEVES',
       'SALES(DHS)'],
      dtype='object')
```

In [53]:
```
Analysing_sales_after_drop.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   DAYS          50 non-null     int64
 1   WOODY         50 non-null     int64
 2   CRESTED VEST  50 non-null     int64
 3   T-SHIRT       50 non-null     int64
 4   LONG SLEEVES  50 non-null     int64
 5   SALES(DHS)    50 non-null     int64
dtypes: int64(6)
memory usage: 2.5 KB
```

In [54]:
```
Analysing_sales_after_drop.dtypes
```

Out[54]:
```
DAYS            int64
WOODY           int64
CRESTED VEST    int64
T-SHIRT         int64
LONG SLEEVES    int64
SALES(DHS)      int64
dtype: object
```

In [57]:
```
x_label = Analysing_sales_after_drop[['DAYS', 'WOODY', 'CRESTED VEST', 'T-SHIRT',
```

In [58]:
```
y_sales = Analysing_sales_after_drop['SALES(DHS)']
```

In [60]:
```
x_label.shape
```

Out[60]:
```
(50, 5)
```

In [61]:
```
y_sales.shape
```

Out[61]:
```
(50,)
```

In [62]:
```
from sklearn.model_selection import train_test_split
```

In [64]:
```
x_train,x_test,y_train,y_test= train_test_split(x_label,y_sales,random_state=1)
```

In [65]:
```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(37, 5)
(13, 5)
(37,)
(13,)
```

```
#LINEAR REGRESSION MODEL
```

In [66]:
```
from sklearn.linear_model import LinearRegression
linreg= LinearRegression()
linreg.fit(x_train,y_train)
```

Out[66]:
```
LinearRegression()
```

In [67]:
```
print(linreg.intercept_)
print(linreg.coef_)
```

```
1.3642420526593924e-12
[1. 1. 1. 1. 1.]
```

In [68]:
```
y_prediction=linreg.predict(x_test)
```

In [69]:
```
y_prediction
```

Out[69]:
```
array([2317., 3167., 2689., 2607., 2048., 3064., 3247., 2407., 1853.,
       2444., 2587., 2374., 2822.])
```

```
#to check accuracy of the prediction
```

In [71]:
```
from sklearn import metrics
import numpy as np
```

In [72]:
```
print(np.sqrt(metrics.mean_squared_error(y_test,y_prediction)))
```

```
7.461609622099486e-13
```

In [73]:
```
print('True',y_test.values[0:10])
print()
print('prediction',y_prediction[0:10])
```

```
True [2317 3167 2689 2607 2048 3064 3247 2407 1853 2444]

prediction [2317. 3167. 2689. 2607. 2048. 3064. 3247. 2407. 1853. 2444.]
```

In [ ]: